MICRO-35 Tutorial

An Introduction to Network Processor Research & Design

Patrick Crowley
University of Washington

pcrowley@cs.washington.edu
http://www.cs.washington.edu/homes/pcrowley

MICRO-35
November 19, 2002
Istanbul, Turkey

An Introduction to Network Processor Research and Design

Patrick Crowley
University of Washington
pcrowley@cs.washington.edu
http://www.cs.washington.edu/homes/pcrowley

Micro-35
November 19, 2002
Istanbul, Turkey

Tutorial Agenda

2:00	**Part 1**	Welcome, Intro & History
2:30	**Part 2**	Design Issues & Challenges
3:00		
3:30	**Break**	
4:00	**Part 3**	Products & Platforms
4:30		*Raj Yavatkar, Intel Corp.*
5:00	**Part 4**	People, Projects and Forums
5:30	**Part 5**	Resources for NP R&D
6:00		Conclude

Introduction

- My view of the audience:
  - People interested in NP research and design
- Goal:
  - Help you get NP R&D started
- Method (& Outline):
  - Intro to NP systems
  - Design issues & challenges
  - Current work
  - Resources

Part 1: Introduction

The **purpose** of Part 1 is to provide technical background for the design issues in Part 2.

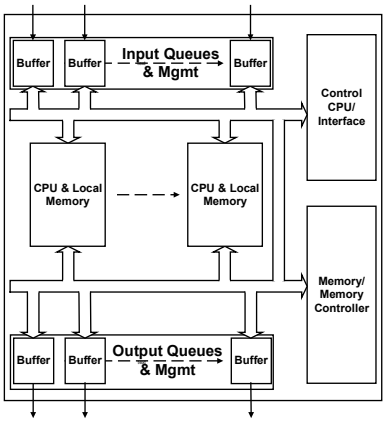
a) Introduction to NP Systems

b) Workloads

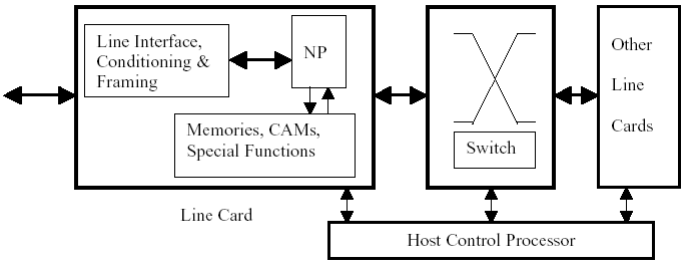
c) Network Processor History

Cut to the Chase: Introduction to NP Systems

- NP system ≥ highly integrated computer
- Packets:
  - Arrive
  - Get processed
  - Depart

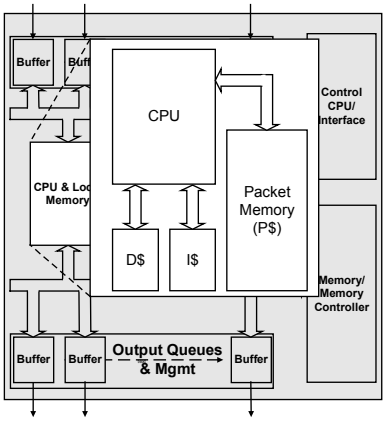


Router Organization



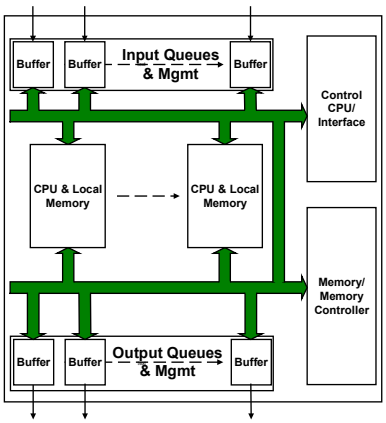
Design Issue: Processor Organization

- 'Do it in software'
- Decisions:
  - Instruction Set
  - High-level architecture
  - Memory & I/O Integration
  - Programming model



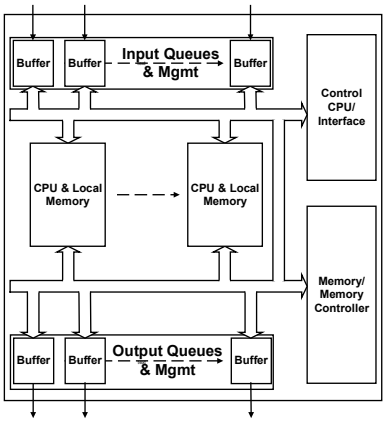
Design Issue: Memory & I/O Path Organization

- As usual, the real problem.
- Decisions:
  - Uniform memory?
  - Distributed memory?
  - Interconnect technology?

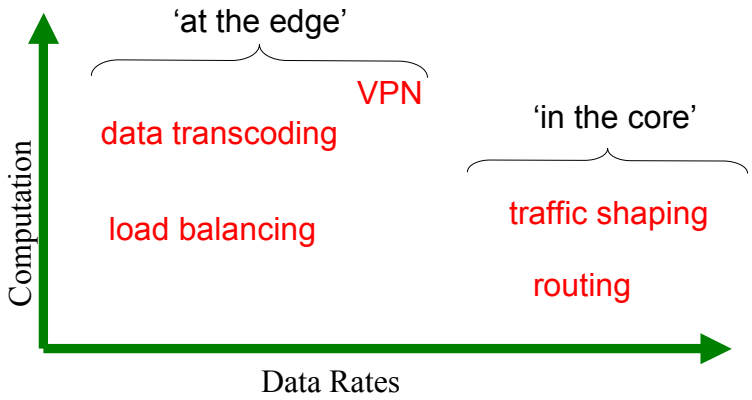


Design Issue: Understanding Workloads

- What will the packets look like?
- What exactly do we do with them?
- How does performance depend on these factors?



Workloads = Traffic + Programs

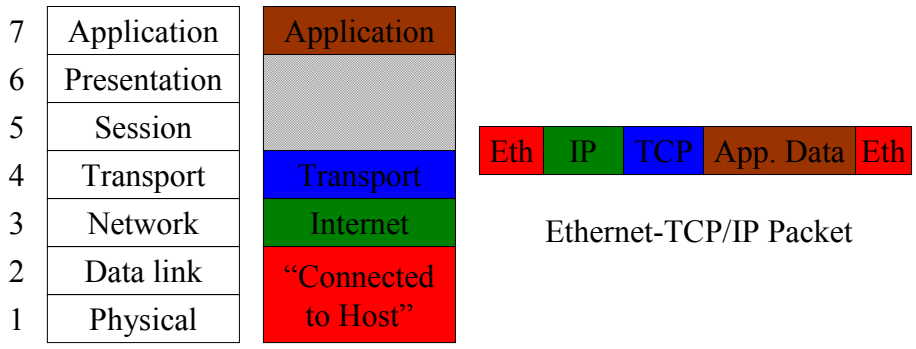


- Range of computational intensity, speeds
- Line rates are increasing everywhere
- **Computation is generally traffic dependent**

Design Issue: Software

- Building a system is no guarantee that you can program it easily:
  - Heterogeneous compute resources
  - Non-uniform memory organization
  - Real-time constraints

Packets & Protocol Layers



OSI TCP/IP
Layered Network Models

Idea: only neighboring layers communicate

Packet Handling

Stage	Description
Media Access Control	Low-level link protocols
Framing/SAR	Handling fragmented packets
Classification	Identifying the packet
Forwarding	Finding the next hop
Modification	Apply transformation
Traffic Management	Schedule transmission

What about policing?

Characteristics of Network Processing Applications

- Packet coverage:
  - Header only, or Header+Payload
- Packet inspection:
  - Is the data location known/static?
  - Reassemble all packets?
- How much state is maintained between packets?
  - Are we counting?
  - Are we basing decisions on dynamic state?
- Traditional distinction: control vs. data plane

Tasks & Services

Applications	Description
Packet Classification/Filtering	Claim/forward/drop decisions, statistics gathering, and firewalling.
IP Packet Forwarding	Forward IP packets based on routing information.
Network Address Translation	Translate between globally routable and private IP packets. Useful for IP masquerading, virtual web server, etc.
TCP connection management	Traffic shaping within the network to reduce congestion.
TCP/IP	Offload TCP/IP processing from Internet/Web servers.
Web Switching	Web load balancing and proxy cache monitoring.
Virtual Private Network (VPN) IP Security (IPSec)	Encryption (DES) and Authentication (MD5)
Data Transcoding	Converting a multimedia data stream from one format to another within the network.
Duplicate Data Suppression	Reduce superfluous duplicate data transmission over high cost links.

Kernels

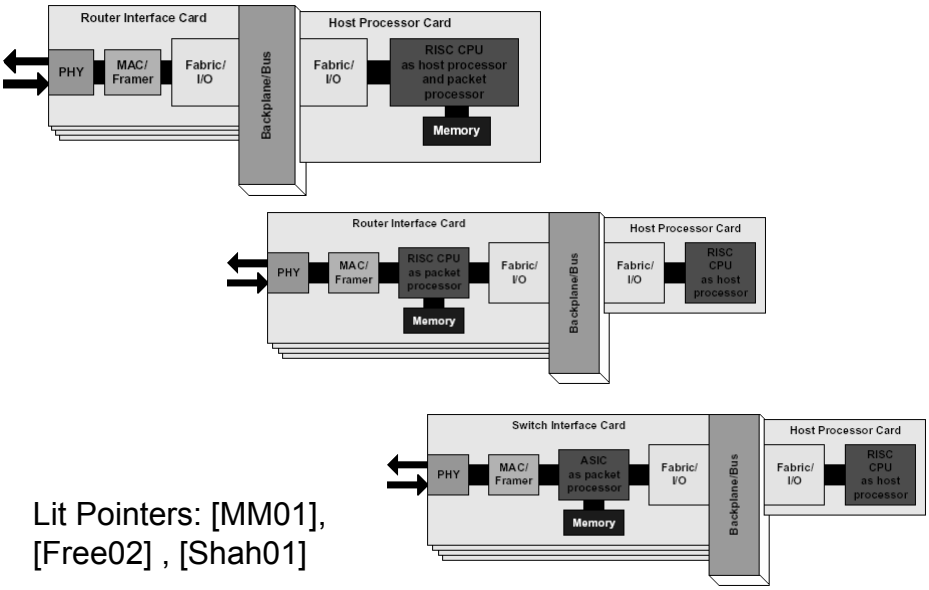
Application	Insts Executed per Message	Loads/Stores (%)	Ctrl Flow (%)	Other (%)
IP forward	~200	25.4	12.7	61.9
MD5	~2000	10.7	2.8	86.5
3DES	~40000	17.8	1.2	81.0

Why Network Processors?

- Arguments:
  - More flexible than ASICs
  - Cheaper than general-purpose processors
  - Better performance than general-purpose processors
  - Software-based functionality provides:
    - Faster time to market
    - Ability to 'fix it later'

Lit Pointers: [AweyaX] , [Free02]

Router History



Lit Pointers: [MM01], [Free02] , [Shah01]

NP History

- Pioneered by MMC Networks
- 30+ startup companies followed
- Lots of acquisitions & big players
  - Intel
  - Motorola
  - IBM
- Lots of attrition

What I mean by *Network Processor*

- Any device that executes programs to handle packets in a data network.
- Examples:
  - processors on router line cards
  - processors in network access equipment

Part 2: Design Issues & Challenges

The **purpose** of Part 2 is to introduce the major technical issues involved in the design and use of network processing systems.

Design Issues:

a) Organizing processor resources
b) Organizing Memory & I/O
c) Instruction Set Architecture
d) Meeting Performance Requirements
e) Writing the Software

Design Issue: Organizing Processor Resources

- Design decisions:
  - High-level organization
  - Instruction set architecture (ISA) and microarchitecture
  - Memory and I/O integration
- Interestingly, today's commercial NPs:
  - Are chip multiprocessors
  - Are multithreaded
  - Exploit little instruction-level parallelism (ILP)
  - Have no caches
  - Are micro-programmed

Question: Why not a Pentium 4?

Not ready to answer the question.

Architectural Comparisons

Consider these high-level organizations:

a) Aggressive superscalar
b) Fine-grained multithreaded
c) Chip multiprocessor
d) Simultaneous multithreaded

Lit Pointers: [CFBB00a], [CFB00c]

Methodology

Applications:

1. Forwarding: IP Forward
2. Authentication: MD5
3. Encryption: 3DES
4. Web balancing: HTTPMON

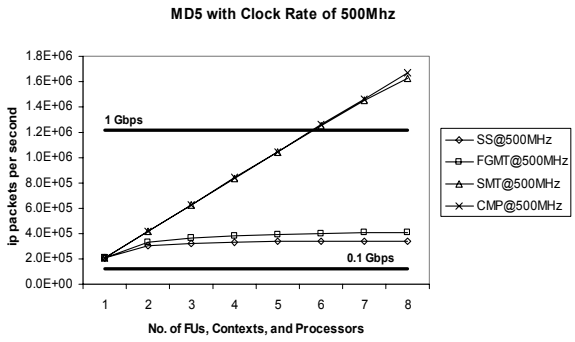
Architectures:

1. Aggressive Superscalar (SS)
2. Fine-grained Multithreaded Processor (FGMT)
3. Chip Multiprocessor (CMP)
4. Simultaneous Multithreaded Processor (SMT)

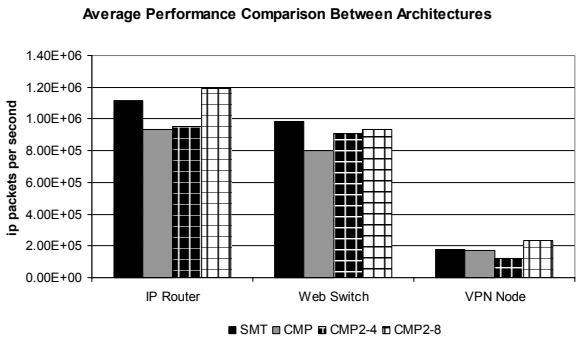
Conclusions:

1. Workloads have little ILP
2. Need to exploit *packet-level parallelism*
3. CMP and SMT do just that.

Standalone Application Performance



SMT vs. CMP2-8

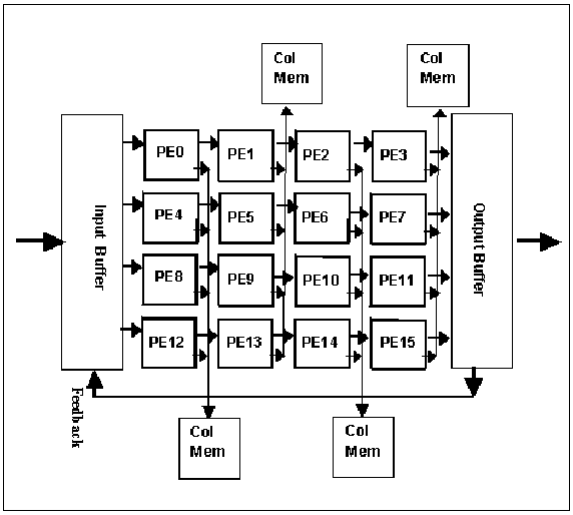


- Adding to cores to CMP2 helps
- So might a multithreaded/smarter OS

Results

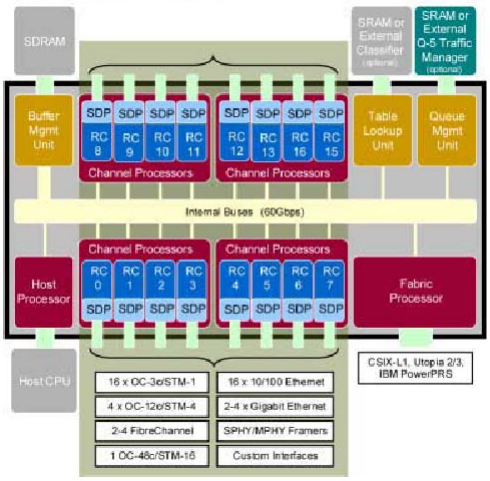
- Systems must support some form of concurrent packet-level parallelism.
  - e.g., threads are a natural mechanism
- OS/Classifier can easily become the bottleneck
- SMT and CMP are nearly equivalent, with SMT always coming out ahead

Example: Cisco ToasterII



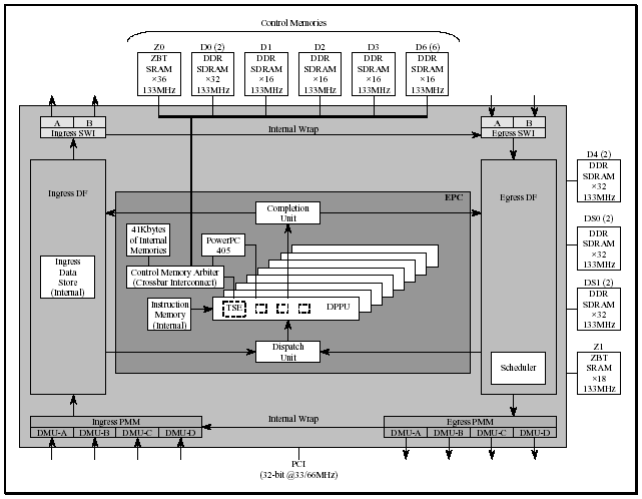
- Each core is a 4 wide VLIW [Marshall02]

Example: Motorola C-5



Lit Pointer: [SJ02]

Example: IBM PowerNP



Lit Pointer: [WL02]

Challenge: Handling Power

For core Internet routers, line density is the principal concern.

Power dissipation is key:

- Each line card has a power budget
- Each line card has a space budget
  - Not much room for heat sinks & fans

Need power efficient designs!

- Possibilities: vectors and stream processors provide lots of computational throughput efficiently

Challenge: Intelligent Design

Given:

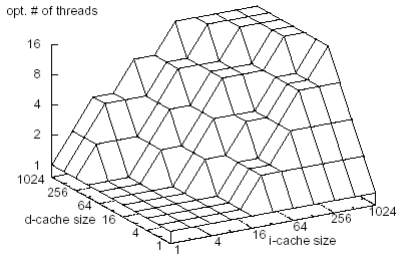
- A selection of programs
- A target network link speed
- A number of network links

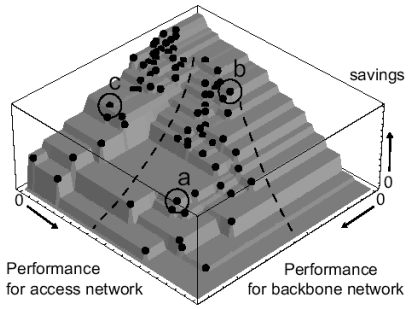
Provide the 'best' design for the processor, where 'best' means:

- Least area
- Least power
- Most performance
- Etc.

Lit Pointer: [TCG02], [FW02]

Examples





- Specific design issue[FW02] & Cost/Benefit Analysis[TCG02]

Design Issue: Memory & I/O Organization

- Must accommodate:
  - Packets flowing through the system
  - Access to program data
  - Sharing between processors and stages of computation
- Provide this flexibly and efficiently

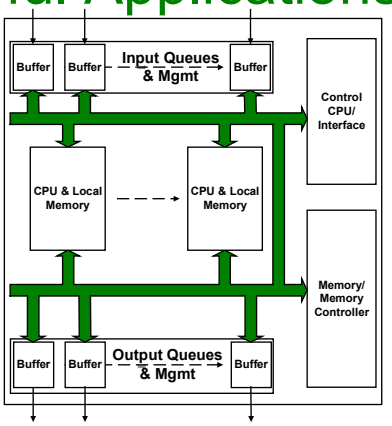
Challenge: Stateful Applications

Example:

Bandwidth allocation

a) 50% to web traffic

b) 50% to UDP traffic



Key: *Forwarding decisions depend on shared state!*

Lit Pointer: [SIP01]

Challenge: Really Fast Networks

a) Network standard: OC-768
   a) OC – optical carrier, i.e., optical fiber
   b) 40 Gbps: 1 OC = 51.85 Mbps
   c) Uses dense wavelength division multiplexing (DWDM)
   d) *Not* cutting edge technology

b) This means:
   a) 78 million 64B packets/s
   b) ~12ns between 64B packet arrivals

Does it make sense to talk about processors and DRAM at these granularities?

Design Issue: Meeting Performance Requirements

Take on the perspective of the user of NPs; the system builder.

- We want to:
  - provide basic networking functionality,
  - *plus* some new feature that customers will pay for.
- **Key question**: Can our system provide basic functionality and implement random feature X at sufficient performance levels?

Challenge: Characterizing Workloads

- **Workloads = Programs + Traffic**
- You can choose a suite of programs
  - *And hope they resemble future programs*
- You can choose a (statistical) traffic model
  - *And hope it resembles your traffic*
- Benchmarks are hard.

Lit Pointers: [Cruz91] , [CB02], [CHY02], [TKS02], [WF00], [MMH01]

Challenge: Average-case vs. Worst-case Performance

- Average case analysis implies some expected traffic model.
- Traffic:
  - Is hard to accurately describe
  - Can vary widely
- Thus: worst-case (or traffic independent) performance is the stable maximum
- Especially for differentiated service routers

Lit Pointer: [KLS98]

Design Issue: Writing the Software

The whole point was to 'do it in software'

- But, our system has:
  a) Heterogeneous compute resources
  b) Non-uniform memory
  c) Multiple interacting threads of execution
  d) Real-time constraints

Challenge: Making use of Resources

- **Goal**: for NPs to be more like general-purpose machines than DSPs.
- Problems:
  - How do programmers use special instructions and hardware assists?
  - Can compilers do it, or is it all hand-coded?

Lit Pointer: [WL02]

Challenge: Writing (Correct) Multithreaded Programs

- If NPs are multi-threaded, then multithreaded programs must be written!
- This means:
  - Managing access to shared state
  - Scheduling policy that ensures correctness
    - Deadlock? Livelock?
- Writing good, correct **single**-threaded programs is hard.

Challenge:Functional & Temporal Correctness

- Stable systems must meet real-time constraints.
  - The current batch of packets must at least be classified before the next arrives
- Can we verify
  - Functional correctness?
  - Temporal correctness?
- Who has experience writing temporally correct multithreaded programs?
- Note: The real-time constraint explains the lack of caches in NPs.

Challenge: Locality & Speculation

- High-performance architectures rely heavily on locality & speculation.
  - Caches, branch prediction, prefetching, …
- Average case improvements justify any non-determinism. *Amdahl's Law.*
- But, what if:
  - You have no average case, and
  - You need good worst-case performance?
- Are locality & speculation applicable?

Question: Why not a Pentium 4?

Answer:

- P4 exploits ILP, not thread-level parallelism
- P4 has a different power budget
- P4 provides non-deterministic performance
  - i.e., hard to make real-time 'guarantees'

Question #2: *What will the answer be in 5 years?*

Summary

- NP system design permits much exploration
  - Parallel and multithreaded architectures
  - Non-standard memory and data paths
  - Worst-case vs. average case emphasis
- Challenges abound

Part 3: Products & Platforms

The **purpose** of Part 3 is to introduce a commercial network processor and network processing platform.

- Raj Yavatkar
  - Chief Software Architect, Intel IXA Architecture Group

This Slide Intentionally Left Blank

Part 4: People, Projects and Forums

The **purpose** of Part 4 is to introduce relevant research projects and forums.

DISCLAIMER: not exhaustive, not perfect,…

- Projects
  - Academia
  - Industrial Research Labs
- Forums

Benchmarking

- CommBench
  - Washington U. in St. Louis
  - `http://ccrc.wustl.edu/~jbf/`
- NetBench
  - UCLA, Bill Mangione-Smith
  - CARES Project
    - `http://www.icsl.ucla.edu/~billms/`
- Berkeley Effort
  - Affiliated with MESCAL project
  - `http://www.gigascale.org/mescal/`
  - Kirk Keutzer

Multiple Projects

- University of Washington
  - Jean-Loup Baer
  - http://www.cs.washington.edu/research/netproc
  - Architectures, Memory Systems, Modeling, Analysis
- Washington University in St. Louis/UMass
  - Mark Franklin & Tilman Wolf
  - http://ccrc.wustl.edu/~jbf/
  - http://www.ecs.umass.edu/ece/wolf/
  - Architectures, Modeling, Analysis, Design

Compilers

- University of Dortmund
  - Jens Wagner
  - http://ls12-www.cs.uni-dortmund.de/~wagner/
  - Backend support for NP instructions

Lookup & Classification

- George Varghese, UCSD
  - http://www.cs.ucsd.edu/users/varghese/
- Nick McKeown, Stanford
  - http://klamath.stanford.edu/~nickm/
  - Also: switch design, memory architectures, scheduling

Operating/Extensible Systems

- Extensible routers, Princeton
  - http://www.cs.princeton.edu/nsg/router.html
  - Larry Peterson
- Spawning Networks, Columbia
  - http://www.comet.columbia.edu/genesis/
  - Andrew Campbell
- Click Modular Router, MIT/ICSI Center for Internet Research
  - http://www.pdos.lcs.mit.edu/click/
  - Kaashoek & Kohler
- Spine, Washington
  - http://www.cs.washington.edu/homes/mef/
  - Bershad & Fiuczynski

Network Test Beds

- Netbed
  - http://www.emulab.net/
  - University of Utah
  - Jay Lepreau
- PlanetLab
  - http://www.cs.princeton.edu/nsg/planetlab/
  - Princeton & others

Industrial Research Efforts

- Bell Labs (Stiliadis)
- Intel Labs
- Nokia
- IBM
- Infineon
- Many others…

Forums: Workshops & Conferences

- Workshop on Network Processors
  - Feb 9, HPCA, Anaheim, CA
  - `http://www.cs.washington.edu/NP2`
  - `http://www.cs.washington.edu/NP1`
- HotChips/HotInterconnects
- Have solicited NP papers:
  - ISCA, ASPLOS, MICRO, HPCA, ICS, etc.
- Industry conferences: NP East & West
  - `http://www.networkprocessors.com`

Forums: Journals

Recent NP-related Special Issues

- IEEE Network
  - `http://www.comsoc.org/pubs/net/ntwrk/special.html`
- Software – Practice & Experience (SPE)
  - `http://www.interscience.wiley.com/jpages/0038-0644/`

Part 5: Resources for NP R&D

The **purpose** of Part 5 is to introduce resources for NP research and development.

- Literature
- Software & Tools
- Equipment & Funding
  - Commercial
  - Governmental

Network Processor Design

- Network Processor Design: Principles & Practices
  - Patrick Crowley, Mark A. Franklin, Haldun Hadimioglu, Peter Z. Onufryk
  - Inspired by NP1
  - From Morgan Kaufmann Publishers,
    - `http://www.mkp.com`
- Contents
  - Technical editors' introduction
  - 7 research papers
  - Market overview
  - 7 Commercial product descriptions
    - Intel, Cisco, PMC-Sierra, IBM, Agere, Transwitch, Motorola

Intel Press

- IXP1200 Programming
  - Erik J. Johnson, Aaron R. Kunze
- Intel Internet Exchange Architecture and Applications: A Practical Guide to Intel's Network Processors
  - Bill Carlson

Networking References

- Interconnections: Bridges, Routers, Switches, and Internetworking Protocols
  - Radia Perlman
- Computer Networks
  - Andrew S. Tanenbaum
- Computer Networks: A Systems Approach
  - Davie & Peterson

Software

- Benchmarks:
  - CommBench
  - NetBench
  - EEMBC
  - NP Forum (?)
    - `http://www.npforum.org`
- Networking Software
  - GNU Zebra, `http://www.zebra.org`
  - Click Modular Router

Tools, Traces & Route Tables

- National Laboratory for Applied Network Research (NLANR)
  - `http://www.nlanr.net`
- Cooperative Association for Internet Data Analysis (CAIDA)
  - `http://www.caida.org`

Intel IXA Educational Program

- Funding and equipment available for IXA-related research and education
- Web sites
  - http://intel.com/research/university/comm/
  - http://www.ixaedu.com

NSF Awards

- Directorate for Computer & Information Science & Engineering (CISE)
- Division of Advanced Networking Infrastructure & Research (ANIR)
  - http://www.cise.nsf.gov/div/anir/index.html

DARPA Awards

- Advanced Technology Office (ATO) Programs
  - http://www.darpa.mil/ato/programs.htm
- Information Processing Technology Office (IPTO)
  - http://www.darpa.mil/ipto/research/index.html

Where to Go From Here

1. Read the literature
2. (Attend NP2 in Anaheim)
3. Talk to companies
4. Choose a problem to solve

Bibliography

[CFHO02] Patrick Crowley, Mark A. Franklin, Haldun Hadimioglu & Peter Z. Onufryk. "Chapter 1: Network Processors: An Introduction to Design Issues" in *Network Processor Design: Issues and Practices.* Morgan Kaufmann Publishers, San Francisco, CA, 2002.

[Free02] John Freeman. "Chapter 9: An Industry Analyst's Perspective on Network Processors" in *Network Processor Design: Issues and Practices.* Morgan Kaufmann Publishers, San Francisco, CA, 2002.

[CFBB00a] P. Crowley, M.E. Fiuczynski, J.-L. Baer, & B. N. Bershad, "Characterizing processor architectures for programmable network interfaces," in *Proceedings of the 2000 International Conference on Supercomputing,* May 2000.

[CFBB00b] P. Crowley, M.E. Fiuczynski, J.-L. Baer, & B. N. Bershad, "Chapter 7: Workloads for Programmable Network Interfaces" in *Workload Characterization for Computer System Design*, Kluwer Academic Publishers, 2000.

[CHY02] P. Chandra, F. Hady, R. Yavatkar, T. Bock, M. Cabot & P. Mathew. "Chapter 2: Benchmarking Network Processors" in *Network Processor Design: Issues and Practices.* Morgan Kaufmann Publishers, San Francisco, CA, 2002.

Bibliography

[TKS02] Mel Tsai, Chidamber Kulkarni, Niraj Shah, Kurt Keutzer and Christian Sauer. "Chapter 7: A Benchmarking Methodology for Network Processors" in *Network Processor Design: Issues and Practices.* Morgan Kaufmann Publishers, San Francisco, CA, 2002.

[WF00] Tilman Wolf & Mark Franklin, "CommBench – A Telecommunications Benchmark for Network Processors," *IEEE International Symposium on Performance Analysis of Systems and Software,* Austin, TX, April 2000, pp. 154-162.

[MMH01] G. Memik, B. Mangione-Smith & W. Hu, "NetBench: A Benchmarking Suite for Network Processors," *International Conference on Computer-Aided Design,* Nov 2001.

[AweyaX] James Aweya, "IP Router Architectures: An Overview," Unpublished manuscript. *On the web*: http://citeseer.nj.nec.com/aweya99ip.html.

[MM01] Bill Mangione-Smith & Gokhan Memik. "Network Processor Technologies," MICRO-34 Tutorial Slides.

[Shah01] Niraj Shah. "Understanding Network Processors," Master's thesis, University of California, Berkeley, September, 2001.

Bibliography

[CFB00c] P. Crowley, M.E. Fiuczynski, & J.-L. Baer, "On the Performance of Multithreaded Architectures for Network Processors," UW Technical Report 2000-10-1.

[TCG02] Lothar Thiele, Samarjit Chakraborty, Matthias Gries & Simon Kunzli. "Chapter 4: Design Space Exploration of Network Processor Architectures" in *Network Processor Design: Issues and Practices.* Morgan Kaufmann Publishers, San Francisco, CA, 2002.

[FW02] Mark A. Franklin & Tilman Wolf. "Chapter 6: A Network Processor Performance and Design Model with Benchmark Parameterization" in *Network Processor Design: Issues and Practices.* Morgan Kaufmann Publishers, San Francisco, CA, 2002.

[SIP01] Devavrat Shah, Sundar Iyer, Balaji Prabhakar, and Nick McKeown. "Analysis of a Statistics Counter Architecture," Hot Interconnects, Stanford, August 2001.

[Cruz91] R. Cruz, "A calculus for network delay," *IEEE Trans. On Information Theory,* 37(1):114-141, 1991.

Bibliography

[CB02] Patrick Crowley & Jean-Loup Baer. "Chapter 8: A Modeling Framework for Network Processor Systems" in *Network Processor Design: Issues and Practices.* Morgan Kaufmann Publishers, San Francisco, CA, 2002.

[KLS98] V.P. Kumar and T.V. Lakshman and D. Stiliadis, "Beyond Best-Effort: Gigabit Routers for Tomorrow's Internet," in *IEEE Communications Magazine* , May 1998.

[WL02] Jens Wagner & Rainer Leupers. "Chapter 5: Compiler Backend Optimizations for Network Processors with Bit Packet Addressing," in *Network Processor Design: Issues and Practices.* Morgan Kaufmann Publishers, San Francisco, CA, 2002.

[Marshall02] John Marshall. "Chapter 11: Cisco Systems – Toaster2," in *Network Processor Design: Issues and Practices.* Morgan Kaufmann Publishers, San Francisco, CA, 2002.

Bibliography

[SJ02] Eran Cohen Strod & Patricia Johnson. "Chapter 14: Motorola – C-5e Network Processor," in *Network Processor Design: Issues and Practices*. Morgan Kaufmann Publishers, San Francisco, CA, 2002.

[WL02] Mohammad Peyravian, Jean Calvignac & Ravi Sabhikhi. "Chapter 12: IBM – PowerNP Network Processor," in *Network Processor Design: Issues and Practices*. Morgan Kaufmann Publishers, San Francisco, CA, 2002.